

WE CLAIM:

1 1. A method for handing Object Request Broker (ORB) communications for a system
2 designed in the Common Object Request Broker Architecture (CORBA), comprising the steps of:
3 creating a plug-in component for a desired client/server application program protocol
4 selected from multiple and diverse communications protocols, said component including a protocol
5 adaptor/connector connecting the ORB and client server applications;

6 structuring a base communications protocol for the ORB to accept said plug-in
7 components so as to translate the client/server application protocol to the General Inter-ORB
8 Protocol (GIOP) of CORBA; and

9 9 combining the plug-in component and the base communications protocol so that the
10 combination ORB handles communications from the client/server application in the desired
11 application protocol.

12 2. The method for handling ORB communications as set forth in claim 1, further
13 including the step of:

14 3 registering at least one other plug-in component for a desired communications
15 protocol out of a plurality of protocols; and
16 5 substituting one of the plug-in components in the combination to cause the system
17 to communicate in the protocol related to the substituted protocol.

18 3. The method for handling ORB communications as set forth in claim 2 wherein the
19 plug-in component includes a scheduler and the schedules changes when the other plug-in is

3 substituted.

1 4. The method for handling ORB communications as set forth in claim 1, wherein the
2 protocol adaptor/connector defines a connector interface which includes a protocol connector.

1 5. The method for handling ORB communications as set forth in claim 1, wherein the
2 protocol adaptor/connector defines an adapter interface which includes a Listener and a dispatcher.

1 6. Apparatus for handing Object Request Broker (ORB) communications for a system
2 designed in the Common Object Request Broker Architecture (CORBA), comprising
3 a plug-in component for a desired protocol related to one of a multiple of diverse
4 communications protocols; and

5 a base communications protocol component for the ORB, said base protocol being
6 adapted to accept said plug-in component so as to translate the internal CORBA structure to a
7 desired communications protocol.

1 7. An article of manufacture having computer readable program means embodied therein
2 for causing Object Request Broker (ORB) communications for a system designed in the Common
3 Object Request Broker Architecture (CORBA) to operate with a desired protocol related to one of
4 a multiple diverse communications protocols, comprising
5 a plug-in component for a desired client/server application program protocol selected
6 from multiple and diverse communications protocols, said component including a protocol

7 adaptor/connector connecting the ORB and client server applications;
8 a base communications protocol for the ORB to accept said plug-in components so
9 as to translate the client/server application protocol to the General Inter-ORB Protocol (GIOP) of
10 CORBA; and
11 wherein the plug-in component and the base communications protocol are combined
12 so that the combination ORB handles communications from the client/server application in the
13 desired application protocol.

1 8. The article of manufacture as set forth in claim 7, further comprising:
2 at least one other plug-in component for a desired communications protocol out of
3 a plurality of protocols, said other plug-in component being registered; and
4 means for substituting one of the other plug-in components in the combination to
5 cause the system to communicate in the protocol related to the substituted protocol.

1 9. The article of manufacture as set forth in claim 8 wherein the plug-in component
2 includes a scheduler and the schedules changes when the other plug-in is substituted.

1 10. The article of manufacture as set forth in claim 7, wherein the protocol
2 adaptor/connector defines a connector interface which includes a protocol connector.

1 11. The article of manufacture as set forth in claim 7, wherein the protocol
2 adaptor/connector defines an adapter interface which includes a Listener and a dispatcher.

1 12. A method for handing requests for service over a communications system designed
2 with multiple protocol levels, comprising the steps of:
3 at a client processor:
4 providing at least one protocol connector;
5 using a client stub component to provide a plug-in protocol connector component for
6 at least one desired client/server application program protocol selected from multiple and diverse
7 communications protocols;
8 using a client connection manager component to establishing a connection between
9 one of said protocol connector and said plug-in protocol connector, and at least one protocol
10 connection component; and
11 connecting the protocol connection to a transport connection, said transport
12 connection establishing communications with a server processor; and
13 at said server processor:
14 providing at least one protocol listener;
15 using a server skeleton component to provide a plug-in listener for at least said
16 desired client/server application program protocol;
17 using a server dispatcher component to establishing a connection between one of said
18 listener and and said plug-in listener, and a protocol adapter, said adapter establishing
19 communications with said transport connection of said client.

1 13. The method for handing requests for service over a communications system as set
2 forth in claim 12, further including the step of:

3 using said client stub to provide at least one other plug-in protocol component for a
4 desired communications protocol out of a plurality of protocols; and
5 using said client connection manager to substitute the other plug-in protocol
6 component for connection to said protocol connector.

1 14. The method for handing requests for service over a communications system as set
2 forth in claim 13, wherein the plug-in component includes a scheduler and the schedules changes
3 when the other plug-in is substituted.